Remarks/Arguments

Claim Amendments

All claim amendments and new claims are fully supported by the specification for the instant application.

No new matter has been added.

The Rejection of Claims 1, 4, 8, 15-26, and 33-35 Under 35 U.S.C. §102

The Examiner rejected Claims 1, 4, 8, 15-26, and 33-35 under 35 U.S.C. §102(b) as being anticipated by U.S. Published Patent Application No. 2002/0091908 (Ashida et al.). Applicant respectfully traverses the rejection.

Anticipation requires that all of the elements of the claim be taught within the four corners of a single reference.

A. Claim 1

1. Ashida does not teach the iterative process recited in Claim 1

The Examiner states that paragraph [0036] teaches the automatic iterative storage process of Claim 1, specifically: "speculation model automatically generated by unit from customer data and speculation data, 0036" and "speculation model and data are used to generate speculation results, 0036".

Applicant has amended Claim 1 to recite further limitations that distinguish the claimed invention from Ashida as follows (added verbiage is underlined):

"a plurality of types of item, in respect of which entries may potentially be provided in a spreadsheet to which the spreadsheet system model relates, the types of item including:

at least one first item type wherein first-item associated data is input data input into the computer system; and

at least two putative second item types wherein second-item associated data can be obtained from an operation performed on stored data, associated with at least one of said first or second item types, stored in a first database, and wherein second item types are not input data;

9

inputting said input data into the system;

automatically searching, using a processor for the computer system, the input data for at least one first item type;

automatically storing, using the processor, data associated with said at least one first item type found by the searching step, in the first database,

automatically performing an iterative determining process, using the processor, for determining whether the first database includes one or more prerequisite items necessary to determine each of a number of putative second item types, the iterative determining process comprising performing a plurality of iterations, wherein:

(a) each iteration of the determining process comprises successively automatically reading putative second item types and, for each read putative second item type, determining whether the first database includes one or more prerequisite items, in the form of first item types and/or second item types, necessary to determine that putative second item type, and if the first database does include said one or more prerequisite items sufficient to determine said second item type, automatically storing that second item type in the first database, such that said second item type can be available as a potential prerequisite item for other second item types in subsequent iterations;

(b) wherein the iterative determining process is terminated by a terminating step under the condition that an iteration of the determining process does not result in storage in the first database of a second item type which was not stored in the first database in a previous iteration of the determining process;

(c) wherein the iterative determining process performs repeated iterations according to step (a) indefinitely until the terminating condition of step (b) is met, each of the second and subsequent iterations assessing putative second item types which were assessed in one or more previous iterations as being unable to be determined due to lack of at least one prerequisite item, and re-assessing those putative second item types taking into account the second item types automatically stored into the first database by previous iterations;"

10

2. The cited excerpts of Ashida do not teach the limitations of amended Claim 1:

None of the following excerpts, cited by the Examiner, teach the iterative process of amended Claim 1:

"a unit to generate the model from the user data and a processing unit to output the result which in turn teaches automation"

"speculation model automatically generated by unit from customer data and speculation data"

"the speculation model and data are used to generate speculation results"

The iterative process of the invention as now recited in Claim 1 provides determination of whether prerequisite items in the form of first item types (wherein first-item associated data is input data input into the computer system) or second item types (ie, non-input data) for determining each second item type exist in the first database, adding determined second item types to the first database and repeatedly performing successive iterations until a termination condition is met, the termination condition being that an iteration of the determining process does not result in storage in the first database of a second item which was not stored in the first database in a previous iteration of the determining process. Each second item type which is stored in the first database can be available as a potential prerequisite item for other second item types in subsequent iterations. This can allow an indefinitely large number of iterations to be performed, allowing determination of second item types which have prerequisites which are second item types which have only been generated after a large number of previous iterations.

3. Ashida teaches a linear process, not the iterative process of Claim 1

Amended Claim 1 recites: "such that said second item type can be available as a potential prerequisite item for other second item types in subsequent iterations;"

Thus, amended Claim 1 recites a loop-like iterative process with feedback that automatically continues iterative steps until a predetermined termination condition is met. As further described *infra*, Ashida does not teach such a process.

The feature that second item types determined in an iteration can be available as a potential prerequisite item for other second item types in subsequent iterations provides amended

11

Claim 1 with an automatic iterative "feedback" functionality, as illustrated, for example in Fig. 3 of the application as filed. That is the results (ie, determined second item types) of each iteration, are stored as part of the database of possible prerequisite items, so that further second item types can be determined based on prerequisite items which are non-input second item types; those further (newly determined) second item types are then stored into the first database(ie effectively 'fed back' into the iterative determining process) and in further iterations can be available as prerequisite items allowing further, previously non-determinable, second item types to be determined; those further determined second item types are stored (or fed back) into the first database and can act as prerequisite items for determination of still further second item types in further iterations, and so on, virtually indefinitely, until the termination condition (that an iteration of the determining process does not result in storage in the first database of a second item which was not stored in the first database in a previous iteration of the determining process) is met. (This termination condition is unambiguously defined in amended Claim 1.) The iterative process may be considered as being "loop-like", to the extent that the iterations are repeated until the termination condition is met.

The above limitation provides significant benefits over a more linearly structured approach, for example as taught by Ashida, because it allows determination of second item types which are an indefinite number of processing steps removed from input data. Further, it allows second item types to be considered in any order rather than requiring that they are considered in an order which ensures their prerequisites (if they exist) would be constructed and determined in a "bottom up" or "top down" manner from input data. For example, in a very simple example based on Example 1 presented in the specification as filed, there are five items: revenue (Rev), expenses (Exp), net cash flow (Cashflow), discount rate (Drate) and net present value (NPV). Rev, Exp and Drate may be regarded as first item types, in terms of the wording of claim 1. Cashflow and NPV are non-input items and may be regarded as second item types in terms of the wording of claim 1. Cashflow can only be determined from Rev and Exp. NPV can only be determined from Cashflow and Drate. Provision of a list of second item types in which NPV appears in the list of second item types to be considered before Cashflow, could create a serious

12

problem for a structured linear analysis, since evaluation of whether the prerequisites for NPV have been determined would conclude that they have not, because when NPV is evaluated the prerequisite Cashflow has not yet been evaluated or determined. Accordingly the result of the evaluation would be that NPV cannot be determined. Clearly this would be an erroneous result, because Cashflow could be subsequently determined, and NPV could be determined thereafter. (Although, in such a simple example, an analysis with a predetermined linear structure could perhaps include some safeguard against such an error, it will be appreciated that this would become progressively more difficult as more second item types, further removed from first item types, are involved.)

However, no such problem is presented in the method recited in amended Claim 1. This is because, (again adopting this simple example) on the first iteration, interrogation of the first database will find that NPV cannot be determined (since a prerequisite, namely Cashflow, is not found in the first database), but then it will be found that Cashflow can be determined (since its prerequisites Rev and Exp can be found in the first database) and Cashflow will then be stored in the first database and be available as a potential prerequisite item for other second item types in subsequent iterations. Then, because at least one second item type (that has not previously been stored in the first database) has been stored in the first database on this iteration, the termination condition has not been met and a further iteration is performed.

On the second iteration, when NPV is evaluated interrogation of the first database will reveal that Cashflow (and Drate) can be found in the first database and it will be established that NPV can be determined. Thus Cashflow and NPV could be entered into the list of second items to be considered in either order without the order affecting the result of the determining process, or the method. Applicant submits that this is in contrast to progressive linear methods, for example, as taught by Ashida, for generating non-input item types which have other non-input item types as prerequisites, since such methods do not provide multiple iterations in which subsequent automatic iterations assess putative second item types which were assessed in one or more previous iterations as being unable to be determined due to lack of at least one prerequisite item, and re-assess those putative second item types indefinitely many times (until they can be

determined, or until the specified termination condition is met) taking into account the second item types automatically stored into the first database by previous iterations, which can act as prerequisite items.

Thus the iterative determining process of amended Claim 1 provides a significant benefit in terms of convenience (and thus economy of time and resources) when entering second item types for assessment into the system. Further, it will be appreciated that multiple other putative second item types each having NPV and/or Cashflow (and/or one or more of the multiple other second item types) as a prerequisite could be provided for assessment for determination, in any order, without affecting the ability of the system to assess whether the second item types can be determined. Further, it will be appreciated that providing a loop-like iterative determining process with a termination condition allows as many iterations as required for a given set of circumstances and first and second item types, in contrast to a more linear system, for example as taught by Ashida, which provides a predetermined number of determining steps.

Applicant appreciates that Ashida appears to disclose generation of non-input items from other non-input items. (For example, as set out in paragraph [0021], speculation results 112, which are non-input data, are generated from customer lists 107 and speculation models 110 which are also non-input data). However, in Ashida generation of each non-input item clearly proceeds, in a **predetermined linear progression**, from construction of previously generated items.

Ashida does not teach the loop-like iterative process with automatic feedback as recited in amended Claim 1.

4. Ashida does not assess putative second item types

Amended Claim 1 recites:

"(a) each iteration of the determining process comprises successively automatically reading putative second item types and, for each read putative second item type, determining whether the first database includes one or more prerequisite items, in the form of first item types and/or second item types, necessary to determine that putative second item type, and if the first database does include said one or more prerequisite items sufficient to determine said second

14

item type, automatically storing that second item type in the first database, such that said second item type can be available as a potential prerequisite item for other second item types in subsequent iterations;"

Ashida, in particular in paragraph [0036], does not teach or suggest an iterative determining process in which subsequent iterations assess putative second item types which were assessed in one or more previous iterations as being unable to be determined due to lack of at least one prerequisite item, and re-assess those putative second item types taking into account the second item types automatically stored into the first database by previous iterations, which can act as prerequisite items, and in which the iterative determining process performs repeated iterations indefinitely until the terminating condition set out in amended Claim 1 is met.

To the contrary, as set out in paragraph [0036] of Ashida the generation of the non-input item types is performed in a **predetermined linear order**: customer data 101 and data definition data 102 are used by a rule generation processing unit 103 to generate characteristic rule sets 104; the rule sets 104, customer data 101 and data definition data 102 are used by segment selector unit 106 to generate speculation data lists or selected customer lists 107 and selected segments 108; the selected segments 108, customer data 101 and data definition information 102 are used by speculation model generation unit 109 to generate speculation models 110; and the selected customer lists 107 and speculation models 110 are used by speculation processing unit 111 to generate speculation results. This is a structured, linear, progressive process and there is no disclosure of an iterative process as set out in Claim 1 as amended (for example, as described *supra*). It is clear that in the methodology of Ashida if, at a given stage in the process, a prerequisite for a given category for some data to be generated does not exist, then (unless generated subsequent in the process by part of the linear progressive process) that prerequisite will never exist without addition of further input data and re-running of the various generating steps.

Thus, Ashida's teachings do not correspond to the automatic iterative determining step recited in amended Claim 1. Since different categories of data are generated at different stages in the process in Ashida, the teaching of Ashida is of a system in which any given assessment is

15

(and need be) considered only once (even though consideration of different model's prerequisites may be consecutively considered). There is no teaching in Ashida of any second or subsequent automatic iteration which attempts to determine an item type which it was previously, unsuccessfully, attempted to determine in a previous iteration, because the process of extracting data from the input (customer) data which is taught by Ashida proceeds along a predetermined linear path, and the iterative process of the present invention is in contrast to such a teaching.

It is clear from, for example, Figure 1 of Ashida, that the process being taught does not include an indefinitely repeated loop-like iterative process (all the arrows define paths *towards* the speculation results). Although Fig. 7 of Ashida does include a loop-like feature, at 704 to 706, it is apparent from a reading of paragraph [0030] that each loop relates to validation of a different speculation model, and that the loop is effectively counting through the different speculation models and verifying each. There is no teaching or suggestion of the iterative determining process defined in amended Claim 1, or the features in the immediately preceding paragraph. Thus the teaching of Ashida is clearly different from the teaching of the present application and does not hint at or suggest the method of amended Claim 1.

5. Ashida does not have the functionality of amended Claim 1

Amended Claim 1 enables effective handling and processing of data items that clearly could not be effectively handled and processed by the method of Ashida. The following example illustrates the novel and useful functioning of amended Claim 1, by way of a less simplified example than that provided above.

In some types of investment banking analysis, a typical problem would involve being given a limited amount of information and being asked to build up the most compete set of hypothetical accounts possible from that limited information, while avoiding the inclusion of unnecessary items (such as null fields) in the final model. (It should be appreciated that such investment banking analysis is frequently concerned with speculations on hypothetical information.)

In the following example, the following list sets out some of the items that it might be desirable to include in a model of a set of forecast pre-tax accounts for a proposed mining

operation. In the following list the labeling of each item (with an upper case letter) is provided for ease of reference. Items for which a value (ie input data) is provided are marked with an asterisk for clarity.

List of items:

Production report

*(A) tonnes run-of-mine production

*(B) recovery rate

(I) tonnes produced

(P) stock movements

(K) tonnes sold

*(D) price per tonne sold

(M) sales revenue

Stockpile report

(K) tonnes sold

*(C) required stockpiles as a percentage of tonnes sold

(Q) stock levels

(P) stock movements

Expenses report

(M) sales revenue

*(E) operating expenses as rule-of-thumb percentage of sales revenue

(N) operating expenses derived from sales revenue

Depreciation report

*(F) capital costs

*(G) depreciation rate

(J) depreciation

Stock valuation report

(N) operating expenses derived from sales revenue

(Q) stock levels

(R) stock valuation

(S) increase in stock value

Profit and loss report

(M) sales revenue

(N) expenses derived from sales revenue

(S) increase in stock value

(J) depreciation

(U) profit before tax

(V) tax

(L) profit/loss

#(T) tax rate

*(H) dividend as a percentage of profit

(O) dividend declared

As set out above, the input data provided relates to: (A) tonnes run-of-mine production, (B) recovery rate, (C) required stockpiles as a percentage of tonnes sold, (D) price per tonne sold, (E) operating expenses provided as a rule-of-thumb percentage of sales revenue, (F) capital costs, (G) depreciation rate, and (H) dividend as a percentage of profit.

The tax rate (T), marked with a hash (#) in the list, is an item for which input data could have been provided, but for which, in the present example, input data has not been provided. (As will be appreciated from the explanation below, provision of input data for the tax rate T would have allowed determination of various other putative items which, in the below example, cannot be determined.)

18

The challenge is to create the most complete set of hypothetical accounts possible, by creating a model with as many of the above items as possible, but which does not include items which cannot be determined from the data provided.

The method of amended Claim 1 takes the approach of modeling "the most complete set of hypothetical accounts possible" iteratively, *with each iteration considering the state after the end of the previous iteration* to see if it can be elaborated further within the constraints of the model specification.

Thus the items that relate to the raw data provided are, in the language of amended Claim 1, first-item types, and the items for which raw data is not provided (and for which the method assesses whether they can be determined) are putative second item types. Thus in accordance with the method set out in claim 1, in this example the items that relate to the raw data provided (the first-item types), namely items (A) to (H), are stored in a first database.

The iterative process begins with a first iteration, in which each putative second item type is assessed by consultation with the database (the first database, in the language of claim 1) to see whether it can be determined, based on whether its prerequisites are found in the first database.

On a first pass of the iterative process, the existence of (A) and (B) could be used to infer the possibility of (I) tonnes produced. The existence of (F) and (G) could be used to infer the possibility of (J) depreciation. (I) and (J) are thus stored in the first database and may be regarded as available prerequisite items in future passes (iterations). None of the other putative second item types can be determined in this iteration. Because at least one second item type (that has not previously been stored in the first database) has been stored in the first database on this iteration the termination condition has not been met, and a further iteration is performed.

On a second pass of the iterative process, the existence of (I) and (C) could be used to infer the possibility of (K) tonnes sold. The existence of (J) could be used to infer the possibility of (L) profit/loss. (K) and (L) are thus stored in the first database and may be regarded as available prerequisite items on future passes (iterations). None of the other putative second item types can be determined in this iteration. Because at least one second item type (that has not

19

previously been stored in the first database) has been stored in the first database on this iteration the termination condition has not been met, and a further iteration is performed.

On a third pass of the iterative process, the existence of (K) and (D) could be used to infer the possibility of (M) sales revenue. The existence of (K) and (E) could be used to infer the possibility of (N) expenses derived from sales. The existence of (L) and (H) could be used to infer the possibility of (O) dividend declared. The existence of (I) and (K) could be used to infer the possibility of (P) stock movements. (M), (N), (L), (O) and (P) are thus stored in the first database and may be regarded as available prerequisite items on future passes (iterations). None of the other putative second item types can be determined in this iteration. Because at least one second item type (that has not previously been stored in the first database) has been stored in the first database on this iteration the termination condition has not been met, and a further iteration is performed.

On a fourth pass of the iterative process, the existence of (M) could be used to infer the possibility of (L), but (L) already exists. The existence of (N) could also be used to infer the possibility of (L), but (L) already exists. The existence of (P) could be used to infer the possibility of (Q) stock levels. (Q) is thus stored in the first database and may be regarded as an available prerequisite item on future passes (iterations). None of the other putative second item types can be determined in this iteration. Because at least one second item type (that has not previously been stored in the first database) has been stored in the first database on this iteration the termination condition has not been met, and a further iteration is performed.

On a fifth pass of the iterative process, the existence of (Q) and (N) could be used to infer the possibility of (R) stock valuation. (R) is thus stored in the first database and may be regarded as an available prerequisite item on future passes (iterations). None of the other putative second item types can be determined in this iteration. Because at least one second item type (that has not previously been stored in the first database) has been stored in the first database on this iteration the termination condition has not been met, and a further iteration is performed.

On a sixth pass of the iterative process, the existence of (R) could be used to infer the possibility of (S) increase in stock value. (S) is thus stored in the first database and may be

regarded as an available prerequisite item on future passes (iterations). None of the other putative second item types can be determined in this iteration. Because at least one second item type (that has not previously been stored in the first database) has been stored in the first database on this iteration the termination condition has not been met, and a further iteration is performed.

On a seventh pass of the iterative process, the existence of (S) could be used to infer the possibility of (L), but (L) already exists. No further items can be inferred from the prerequisites in the database, so no items are added to the database on this pass.

As no further second item types have been added to the database on the seventh pass, in accordance with amended Claim 1, the iterative process terminates since the termination condition has been met (it will not be possible to determine further second item types by further passes/iterations). Further, in accordance with amended Claim 1, the storage of an item type in the first database is an indication that the stored item type may usefully be included in a spreadsheet in accordance with the spreadsheet system model. In this example, this is an indication that the numerical input data provided for items (A) to (H) has allowed generation of useful values for these second item types.

It may be observed that the putative second item types (U) profit before tax and (V) tax are never added to the first database. Had input data for (T) tax rate been provided these items could have been inferred automatically by the model specification and added to the first database. Many other similar examples could be given.

Conversely, had no input data been provided for (C) required stockpiles as a percentage of tonnes sold, then no (K) tonnes sold would have been determined, and added to the first database. As can be appreciated from the example set out above, (K) has been used (directly or indirectly) in the evaluation of (P), (Q), (R) and (S), and hence if input data for (C) had not been provided, (P) stock movements, (Q) stock levels, (R) stock valuation and (S) increase in stock value would not have been able to be determined (and consequently would not have been added to the first database, and would not be indicated as items which could usefully be included in the spreadsheet in accordance with the spreadsheet system model).

Alternatively, had no input data been provided for (E) operating expenses as rule-of-thumb percentage of sales revenue, then no (N) expenses derived from sales would have been added, and hence no (R) stock valuation or (S) increase in stock value - even though input data *had* been provided for (C) required stockpiles as a percentage of tonnes sold.

It will be appreciated that the disclosure of Ashida is fundamentally unsuited to performing an analysis in this manner. The methodology of Ashida would require that all the proposed item types be considered in the particular order which assesses the prerequisites of each given second item type prior to that given second item type, making construction of the assessment system difficult, laborious, inflexible and prone to error. For example, if an additional second item type were to be added into the methodology of Ashida, a detailed analysis of where (or when) to perform an assessment of that second item type would be required, and an error in that analysis would adversely affect assessment. However, in the iterative methodology of the present invention, an additional second item type could be added anywhere in the list of second item types to be assessed.

For example, it may be desirable to allow for the creation of the most complete balance sheet possible from a given set of input data, and to make provision for a new item - (X) ratio of current liabilities to current assets. Item (X) would be required *if and only if* there existed any of trade creditors or general creditors or taxation payable or interest payable or bank overdraft or any of a number of types of provision AND any of cash or interest receivable or trade debtors or other debtors or stock or any of a number of types of consumables. Each of the current liability items and each of the current asset items would themselves depend on a chain of relationships back to possible input data which might or might not be provided in any particular case. It may be readily observed that to make such a modification using a linear process (such as that of Ashida), it would be necessary to consider in advance all of the possible pathways by which that new item could arise from different possible combinations of input items. It would then be necessary to construct the linear process accordingly in order to attempt to generate a value for (X), or evaluate whether (X) can be determined. An error in construction of the linear process

could result in an incorrect conclusion that (X) could not be determined, since a data set or segment required to determine (X) might be determined later, rather than earlier, in the process.

Conversely, it might be decided to make provision for another possible input item (the inventor notes, for example, that a typical problem may involve recovery of two separate minerals – lead and copper - from one stream of ore). It may be readily observed that to make such a modification using a structured linear process, it would be necessary to consider in advance all of the direct and indirect consequences of that change on potential non-input items. In contrast, in both cases the *loop-like iterative process with feedback* of the present invention allows for modification to be made only to immediate dependencies (prerequisite items), since more distant dependencies will be handled automatically by the iterative process itself. Because an indefinite number of iterations are performed (until the termination condition is met) it would not matter if the prerequisites for (X) were presented in a list of second items to be evaluated *after* (X), because (X) would be evaluated again in one or more subsequent iterations.

Several further points may be noted:

- The method of amended Claim 1 avoids creating unnecessary items in the final spreadsheet model. For example, because no tax rate has been included as user data, items relating to the calculation of tax are not created in the spreadsheet. Likewise, because no loan information has been included as user data, items relating to loan drawdown, loan repayment, and interest calculation and payment are not created.

- It may be seen that this process proceeds iteratively, *with each iteration considering the state at the end of the previous iteration*. No user involvement is required, nor would it be useful to have user involvement.

- In contrast, Ashida specifically envisages user interaction [105 of Figure 12] in selecting "speculation models" ("the user 105 independently selects one of the speculation models"). Whether or not that particular embodiment was used, the fact that it is even contemplated indicates that iteration of the type described above is not contemplated.

- The use of an iterative methodology of the type described gives rise to the possibility of a catastrophic infinite loop, as noted in the specification as filed. In order to avoid this a

23

suitable termination condition for the iterative process must be included (as recited, for example, in claim 1). As noted in previous submissions Ashida does not teach any conditions necessary for either normal termination of iteration or for prevention of such catastrophic outcomes. This again emphasizes that Ashida does not teach the iterative process of current claim 1.

The fact that the bulk of the comments in this paper relate to the differences between the iterative process of amended Claim 1 and the linear step-by-step process of Ashida should not be taken as an admission by Applicant that Ashida discloses all the other features of amended Claim 1.

For example, Ashida is directed to extraction of groups of data from input data lists in order to asses correlations between segments of data and certain observed behaviors, rather than to "a computer implemented method for processing data for a spreadsheet system model" per se (and still less to such a method which can provide an indication of whether a putative item type may usefully be included in a spreadsheet in accordance with the spreadsheet system model based on whether that item type is stored in a first database).

Applicant submits that at least the discussed differences between the iterative process of the present invention and the linear step-by-step process of Ashida are sufficient to establish patentability of amended Claim 1, but makes no admission that the other features of amended Claim 1 are disclosed in Ashida.

For all the reasons noted above, Ashida fails to teach each and every element of Claim 1 and Claim 1 is novel with respect to Ashida. Claims 4, 8, 15-26, and 33-37, dependent from Claim 1, also are novel with respect to Ashida.

B. New Claim 38

Ashida fails to teach each and every element of Claim 38; therefore, Claim 38 is novel with respect to Ashida.

Applicant courteously requests that the rejection be removed.

The Rejection of Claims 5-7 and 9-14 Under 35 U.S.C. §103

The Examiner rejected Claims 5-7 and 9-14 under 35 U.S.C. §103(a) as being unpatentable over U.S. Published Patent Application No. 2002/0091908 (Ashida et al.) and in view of U.S. Patent No. 6,438,547 (Mehr et al.). Applicant respectfully traverses the rejection.

Applicant has shown that Ashida fails to teach every element of Claim 1. Nor does Ashida suggest or motivate all the elements of Claim 1. Therefore, Claim 1 is patentable over Ashida. Mehr does not cure the defects of Ashida with respect to Claim 1; therefore, Claim 1 is patentable over the cited references. Claims 5-7 and 9-14, dependent from Claim 1, enjoy the same distinction with respect to the cited references.

Applicant courteously requests that the rejection be removed.

The Rejection of Claims 26-31 Under 35 U.S.C. §103

The Examiner rejected Claims 26-31 under 35 U.S.C. §103(a) as being unpatentable over U.S. Published Patent Application No. 2002/0091908 (Ashida et al.) and in view of U.S. Published Patent Application No. 2002/10049749 (Helgeson et al.). Applicant respectfully traverses the rejection.

Applicant has shown that Claim 1 is patentable over Ashida. Helgeson does not cure the defects of Ashida with respect to Claim 1; therefore, Claim 1 is patentable over the cited references. Claims 26-31, dependent from Claim 1, enjoy the same distinction with respect to Claim 1.

Applicant courteously requests that the rejection be removed.

Conclusion

Applicant respectfully submits that all pending claims are now in condition for allowance, which action is courteously requested.

Respectfully submitted,

/Chester Paul Maliszewski/

Chester Paul Maliszewski
Registration No.51,990
CUSTOMER NO. 24041
Simpson & Simpson, PLLC
5555 Main Street
Williamsville, NY 14221-5406
Telephone No. 716-626-1564

CPM
Dated: September 17, 2009